

AMENDMENTS TO THE CLAIMS

1-15 (Canceled)

16. (Currently Amended) An N+1 parallel program (P-P) module based on a single machine environment, ~~which comprises~~comprising:

a ~~N+1 P-P~~ an N+1 P-P branch programs module, where N is greater or equal to 1, which is run by operating the N+1 P-P branch programs which have an object code independent structures, in the way of time division, for making a transmission and consistency of P-P data in the P-P branch programs under the supports of three classes of sequence-net instructions for reading P-P data, writing P-P data, and making P-P data consistency ~~in said P-P branch programs~~; and

a managing program module, for supporting a suspension status, a ready status, and a running status of the P-P branch programs in response to information from the P-P branch programs,

wherein the classes of the sequence-net instructions for reading P-P data and writing P-P data are only executed by the 1st to Nth P-P branch programs, and the class of the sequence-net instruction for making P-P data consistency is only executed by the N+1th P-P branch program.

~~wherein, the N+1th P-P branch programs module executes a P-P data sequence, which is represented by a data consistency operation.~~

17. (Previously Presented) The N+1 P-P module of claim 16, further comprising a suspension-processing module, for processing the suspension status, the ready status and the running status of the current P-P branch programs.

18. (Previously Presented) The N+1 P-P module of claim 16, further comprising a P-P entering & exit management module, for initializing the P-P for each application and processing an exit of the P-P in response to information from a time-division- managing module.

19. (Cancelled)

20. (Currently Amended) The N+1 P-P module of ~~claim 16~~claim 32, wherein a token consistency operation of the N+1th branch program corresponds to the P-P data, which is made consistency one by one directly by the control of the N+1th program, wherein if the P-P data has been written by other P-P branch program, then the consistency of the P-P data is executed, otherwise the N+1th branch program is in the suspension status, after the execution of the P-P data's consistency, the P-P data is valid when being read by other P-P branch program-is-valid.

21. (Currently Amended) The N+1 P-P module of claim 20, wherein, ~~when child P-Ps are~~when a child P-P is called, a call-permission instruction of the child P-P and the P-P data sequence are sequenced in the N+1th branch program.

22. (Currently Amended) The N+1 P-P module of claim 20, ~~wherein, the N+1 programs include three kinds of P-P instructions for reading the data, writing the data, and~~

~~making the data in consistency respectively, and wherein the three classes of the sequence-net~~
the P-P instructions have the capability for setting, detecting, and processing tokens, and also have the capability for supporting transmissions of P-P data and synchronization of the P-P branch programs.

23. (Currently Amended) An ~~operational-operation~~ method of N+1 parallel program (P-P) programs, where N is greater than or equal to 1, based on single machine environment, ~~which comprises the following procedures~~the method comprising:

initializing the N+1 P-P branch programs,

determining whether a P-P is terminated, and checking an ON/OFF switch flag of each P-P branch program, wherein:

~~if a switch flag~~switch flags of all P-P branch ~~programs is~~programs are OFF, which indicates the P-P is terminated, record for showing the termination of the P-P and the manner for connecting with external programs are processed,

~~once the~~if the switch flag ~~of one of a~~of one of a P-P branch program is ON, checking a suspension status for the branch programs is checked, wherein,

if the check result is that all the P-P branch programs are in a suspension
the suspension status, which indicates that though the execution of some P-P branch programs is not finished, the P-P branch programs can not be executed,
then

finding out reasons that the P-P branch programs can not be executed,

processing an exit of the P-P branch programs based on the reasons,

if the check result is that more than one P-P branch program is in ready status, then ~~enter the~~enters a P-P branch programs queuing ~~process~~module,
queuing module selecting a P-P branch program which is in the ready status for the P-P branch programs queuing process,
selecting a P-P branch program, loading its parameters, and
running the P-P branch program ~~which is selected~~,
wherein the P-P branch programs comprise a P-P data writing subroutine, a P-P data reading subroutine and a P-P data consistency subroutine, wherein the P-P data writing subroutine and the P-P data reading subroutine are only executed by the 1st to Nth P-P branch programs, and the P-P data consistency subroutine is only executed by the N+1th P-P branch program.
~~wherein, the N+1th P-P branch program of P-P module executes a P-P data sequence, which is represented by a data consistency operation.~~

24. (Currently Amended) The N+1 P-P operation method of claim 23, ~~wherein,~~
wherein the initializing ~~procedure step~~ comprises ~~loading the~~ loading of parameters of the P-P branch programs and the and resetting of a flag zone of the P-P branch programs.

25. (Currently Amended) The N+1 P-P operation method of claim 24, wherein, the loading of the parameters of the P-P branch ~~program~~programs comprises:
loading the flag ON for the N+1 P-P branch programs ~~of the P-Ps~~,
setting an entry address for each of the P-P branch programs,
setting~~setting~~ a data initial values for respective registers, and

resetting a P-P flag zone and a data flag of the P-P branch program.

26. (Currently Amended) The N+1 P-P operation method of claim 23, wherein the P-P branch program comprises a subroutine for writing data, which comprises the following ~~procedures~~data writing subroutine comprises the following steps:

~~performing a~~performing the writing operation ~~for a P-P data for the P-P data,~~

checking a flag “had been invalid for consistency” of the P-P data,

if the flag is valid, the status bit ~~of N+1th of the~~ P-P branch program ~~of thein a~~ ready/suspension flag ~~of P-P~~ is changed from the suspension status to the ready status,

if the flag is invalid, no specific operation is executed,

establishing a flag “consistency valid” and allowing a consistency P-P branch program to perform a consistency operation to the P-P data, and

exiting the P-P data writing ~~subroutine for writing data.~~

27. (Currently Amended) The P-P operation method of claim 23, wherein the P-P branch program comprises a ~~consistency data subroutine, which comprises the following~~ ~~procedures~~data consistency subroutine comprises the following steps:

checking a “consistency valid” flag of this P-P data, wherein,

if the “consistency valid” flag of this P-P data is valid, consistency operations of data and token are executed,

checking whether an N bits “had been read invalid” flag of this P-P data is valid,

if the “had been read invalid” flag is valid, then the relevant P-P branch programs are changed to the ready status from the suspension status based on the content of the flag “had been read invalid”,

if the “had been read invalid” flag of this P-P data is invalid, no specific operation is proceeded,

the P-P data consistency subroutine is terminated and returned,

if the “consistency valid” flag of this P-P data is invalid, the P-P branch program is suspended,

setting a “had been invalid in consistency” flag,

proceeding the suspension process of the current ~~N+1th~~-branch program, saving the current running situation status of the ~~N+1th~~-P-P branch program to be used when returning the P-P branch program, and

exiting the P-P branch program, and keeping the P-P branch program in the suspension status and quitting from a returning port, in order to re-select a new P-P branch program.

28. (Currently Amended) The P-P operation method of claim 23, wherein the P-P ~~branch program comprises a read data~~ reading subroutine, which comprises the following ~~procedures~~steps:

~~checking the~~ checking a “read valid” flag of the P-P data,

~~if a read if the~~ “read valid” flag of the P-P data is valid, then

reading the P-P data,

returning ~~the read~~ the P-P data reading subroutine,

if the “read valid” flag of the P-P data is invalid,

~~in accordance with the branch program of the read data subroutine, one of~~
~~N bits “had been read invalid” flag is set as “had been read invalid” respectively,~~
N “had been read invalid” bits are set corresponding to respective P-P
branch programs.

processing the current P-P branch program in the suspension status, saving
the current running ~~situation-status~~ of the P-P branch program so that the current
P-P branch program may be continued from the suspension point, and
exiting the P-P branch program exits and turns and turning to a branch
suspension & exit ~~module port~~ in order to re-select a new P-P branch program.

29. (Currently Amended) A parallel processing (P-P) call-instruction including entry
addresses of N+1 P-P branch programs based on a single machine environment, which
~~comprises~~comprising:

a call-instruction, for calling a P-P entering & exit management module which is used to
process and acquire the entry addresses of the parallel call of the branch programs,

a time-division-managing module for managing operations of multiple branch programs,
wherein the time-division-managing module selects and runs a first branch program from the
multiple branch programs, manages exiting and returning of the first branch program, and selects
and runs a second branch program from the multiple branch programs; the time-division-
managing module returns to the P-P entering & exit management module until all the multiple
branch programs ~~is run~~are run.

wherein the P-P branch programs comprise three classes of sequence-net instructions for writing P-P data, reading P-P data and making P-P data consistency, wherein the classes of the sequence-net instructions for writing P-P data and reading P-P data are only executed by the 1st to Nth P-P branch programs, and the class of the sequence-net instruction for making P-P data consistency is only executed by the N+1th P-P branch program.

wherein, the N+1th P-P branch program of P-P module executes a P-P data sequence, which is represented by a data consistency operation.

30. (Currently Amended) The time-division-managing module of claim 29, wherein the time-division managing module manages a “suspension status”, a “ready status”, and a “running status” of the multiple branch programs in response to return information from the N+1 P-P branch programs in the suspension status.

31. (Currently Amended) A method for managing a parallel processing (P-P) call-instruction and call-permission-instruction based on a single machine environment, which comprisescomprising the following steps:

executing the call-permission-instruction before executing the P-P call-instruction,

locating the call-instructions and the call-permission-instructions in different branch programs respectively, and

sequencing the call-permission-instructions and the P-P data sequence in the N+1th branch program simultaneously,

wherein, the P-P call-instruction includes entry addresses of N+1 P-P branch programs, and

wherein the P-P branch programs comprise three classes of sequence-net instructions for writing P-P data, reading P-P data and making P-P data consistency, wherein the classes of the sequence-net instructions for writing P-P data and reading P-P data are only executed by the 1st to Nth P-P branch programs, and the class of the sequence-net instruction for making P-P data consistency is only executed by the N+1th P-P branch program.

~~wherein, the N+1th P-P branch program of the N+1 P-P branch programs executes a P-P data sequence, which is represented by a data consistency operation.~~

32. (New) The N+1 P-P module of claim 16, wherein the N+1th P-P branch program executes a P-P data sequence, which is represented by a data consistency operation.